UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/705,525	11/10/2003	Attila Barta	RSW920030176US1	5400

51016 7590 06/20/2006

IBM CORP. (RALEIGH SOFTWARE GROUP)
c/o Rudolf O Siegesmund Gordon & Rees, LLP
2100 Ross Avenue
Suite 2600
DALLAS, TX 75201

EXAMINER	
CHEN, QING	
ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 06/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

18

Office Action Summary	**Application No.** 10/705,525	**Applicant(s)** BARTA ET AL.	
	Examiner Qing Chen	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 November 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-40* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 February 2004* is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This is the initial Office action based on the application filed on November 10, 2003 and the preliminary amendment filed on November 28, 2003, which is accepted as being in compliance with 37 CFR § 1.121(b)(1)(ii). **Claims 1-40** are currently pending and have been considered below.

Drawings

2. The replacement drawing was received on February 23, 2004. The drawing is not acceptable because of non-compliance with 37 CFR § 1.121(d). Any changes to an application drawing must be in compliance with 37 CFR § 1.84 and must be submitted on a replacement sheet of drawings, which shall be an attachment to the amendment document and, in the top margin, labeled "Replacement Sheet."

3. Figure 1A should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: "Installation and Removal of Software Components Across Enterprise Resources."

5. The abstract of the disclosure is objected to because it contains a paragraph number. Correction is required. See MPEP § 608.01(b) and 37 CFR § 1.72(b).

6. The disclosure is objected to because of the following informalities:

- The specification contains the following typographical errors:
  - The word "hardware" in the first sentence after "many computer systems a decade ago" should be deleted in page 1, paragraph [3].
  - The word "is" in the sentence "… for an administrator is devise reliable test plans …" should presumably be read "to" in page 2, paragraph [5].
  - The company name "Redhat" should be "Red Hat" in page 3, paragraph [6].
- The specification contains missing patent numbers and filing dates for patents incorporated by reference in paragraphs [26], [28], and [33]. Also, the IBM Disclosure numbers in the respective paragraphs should be deleted.

Appropriate correction is required.

7. The use of the trademarks, such as SUN, JAVA, J2EE, RED HAT, LINUX, and JVM, has been noted in this application. Trademarks should be capitalized wherever they appear (capitalize each letter OR accompany each trademark with an appropriate designation symbol, e.g., ™ or ®) and be accompanied by the generic terminology (use trademarks as adjectives modifying a descriptive noun, e.g., "the LINUX operating system").

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Objections

8. **Claims 20, 21, 39, and 40** are objected to because of the following informalities:

- **Claim 20** contains the following typographical errors:
  - o The word "incompatibilities" is misspelled in the preamble.
  - o The phrase "previously component" should presumably be read "previously installed component."
  - o There should be a comma (,) at the end of the second limitation to separate the "whereby" clause, not a semicolon (;).
- **Claim 21** contains a typographical error: there should be a hyphen (-) between the words "computer" and "readable" in the preamble.
- **Claims 39 and 40** contain a typographical error: the word "compatibility" is misspelled in the limitation body.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

9. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

10. **Claims 1-22, 28, 32, and 34-40** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The language of **Claims 1-19 and 34-40** raises a question as to whether the claims are directed merely to an abstract idea that is not tied to a technological art, environment, or machine, which would result in a substantial practical application producing a useful, concrete, and tangible result to form the basis of statutory subject matter under 35 U.S.C. § 101.

The claimed inventions of Claims 1-19 and 34-40 are not tangible embodied, as they do not require the use of hardware to realize the prescribed functionality.

It is noted that each of the means in **Claims 14-16 and 18-19** can be reasonably interpreted as software alone. Therefore, Claims 14-16 and 18-19 are directed to systems of functional descriptive material *per se*, and hence non-statutory for that reason as well. The claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

It is noted that **Claim 20**, merely claimed as a data structure, contains functional descriptive material *per se*, and therefore, is held non-statutory. Data structures not claimed as embodied in computer-readable media are descriptive material *per se*, and are not statutory because they are not capable of causing functional change in the computer. See, e.g., *Warmerdam*, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure *per se*, held non-statutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention, which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components, which permit the data structure's functionality to be realized, and is thus statutory.

Furthermore, **Claim 17** is directed to a system containing a data structure. When the claimed invention taken as a whole is directed to a mere program listing (i.e., to only its description or expression) without the computer-readable medium needed to realize the computer program's functionality, it is descriptive material *per se*, and hence non-statutory.

The result of **Claims 1, 2, 8, 14, 15, 21, 22, 28, and 39** is directed to the act of "identifying," which does not appear to be a tangible result so as to constitute a practical application of the idea. The act of "identifying" is merely a thought or an abstract idea and does not appear to produce a tangible result even if the step of identification does occur, since the result of that identification is not conveyed in the real world. The result is an identification, which is neither used in a disclosed practical application nor made available for use in a disclosed practical application. It also does not appear that the usefulness of the identification can be realized from the claimed steps to support a disclosed specific, substantial, and credible utility so as to produce a useful result.

In addition, the result of **Claims 12 and 32** is directed to the act of "indicating" and the result of **Claims 34 and 40** is directed to the act of "determining," which also does not appear to be a tangible result so as to constitute a practical application of the ideas. The reasoning is stated above in the aforementioned paragraph and applied in the same manner.

Therefore, the claims do not meet the statutory requirement of 35 U.S.C. § 101, since the claims are not directed to a practical application of the § 101 judicial exception producing a result tied to the physical world.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

12. **Claims 1-4, 6-24, 26-35, and 37-40** are rejected under 35 U.S.C. 102(b) as being anticipated by **Curtis (US 6,442,754).**

As per **Claim 1,** Curtis discloses a method for installing software components, comprising:

A. Identifying components previously installed on a system *(see Column 10, Lines 60-64; and Column 11, Lines 11-20 and 57-61)*;

B. Identifying components to be installed on the system *(see Column 6, Lines 28-40 and 53-58; and Column 11, Lines 20-23)*; and

C. Identifying any potential conflicts between a previously installed component and a component to be installed *(see Column 12, Lines 27-32).*

As per **Claim 2,** Curtis discloses a method for installing software components **as in Claim 1 above,** and further discloses that the identifying components previously installed on a system comprises accessing a semantic model comprising relationships among previously installed components *(see Figure 5; Column 11, Lines 45-52; Column 13, Lines 7-27; and Column 14, Lines 42-45).*

As per **Claim 3,** Curtis discloses a method for installing software components **as in Claim 2 above**, and further discloses that the method comprising updating the semantic model with the identity of newly installed components *(see Column 11, Lines 63-65; Column 12, Lines 66-67; and Column 13, Lines 28-47).*

As per **Claim 4,** Curtis discloses a method for installing software components **as in Claim 1 above**, and further discloses that the method comprising providing a user with a plurality of options if a conflict is identified *(see Column 12, Lines 35-45).*

As per **Claim 6,** Curtis discloses a method for installing software components **as in Claim 4 above**, and further discloses that a second option includes continuing the installation *(see Column 12, Lines 51-53 and 57-59).*

As per **Claim 7,** Curtis discloses a method for installing software components **as in Claim 6 above**, and further discloses that the method including, upon the exercise of the second option, recording an entry in a log indicative of the conflict and of the continuation of installation *(see Figure 2, Element 140; Column 6, Lines 65-67; Column 7, Lines 4-5; and Column 8, Lines 24-29).*

As per **Claim 8,** Curtis discloses a method for installing software components **as in Claim 1 above,** and further discloses that the method comprising:

A. Initiating a removal of a component from the system *(see Figure 6, Element 560; and Column 13, Lines 50-51)*; and

B. Identifying remaining components which depend on the component to be removed *(see Column 13, Lines 59-62).*

As per **Claim 9,** Curtis discloses a method for installing software components **as in Claim 8 above,** and further discloses that the method comprising providing a user with a plurality of options if a dependent remaining component is identified *(see Column 12, Lines 35-45).*

As per **Claim 10,** Curtis discloses a method for installing software components **as in Claim 9 above,** and further discloses that a first includes aborting the removal *(see Figure 6, Element 570; and Column 13, Lines 62-63).*

As per **Claim 11,** Curtis discloses a method for installing software components **as in Claim 9 above,** and further discloses that a second option includes continuing the removal *(see Figure 6, Element 568; and Column 13, Lines 55-56).*

As per **Claim 12,** Curtis discloses a method for installing software components **as in Claim 8 above,** and further discloses that the method comprising:

A. Identifying a first component previously installed on the system which is dependent upon a removed component *(see Column 13, Lines 4-6 and 64-67)*; and

B. Indicating the identity of a second component upon which the first component depends *(see Column 13, Lines 1-4)*.

As per **Claim 13**, Curtis discloses a method for installing software components **as in Claim 12 above**, and further discloses that the method comprising:

A. Installing the second component upon which the first component depends *(see Column 13, Lines 1-4)*; and

B. Creating a dependency link between the first and second components *(see Column 13, Lines 28-47)*.

As per **Claim 14**, the Applicant appears to be attempting to invoke 35 U.S.C. 112, sixth paragraph, since it contains "means-plus-function" limitations. However, the Examiner notes that the only "means" for performing these recited functions in the specification appears to be computer program modules—software *per se*. While the claim meets the first prong of the three-prong analysis used to determine invocation of 35 U.S.C. 112, sixth paragraph, the claim does not meet the other prongs of the three-prong analysis, since no other specific corresponding structure or equivalents thereof are disclosed in the specification. Therefore, the claim limitations are not being treated under 35 U.S.C. 112, sixth paragraph.

Curtis discloses a system for installing software components, comprising:

A. Means for identifying components previously installed on a system *(see Column 10, Lines 60-64; and Column 11, Lines 11-20 and 57-61)*;

B. Means for identifying components to be installed on the system *(see Column 6, Lines 28-40 and 53-58; and Column 11, Lines 20-23)*; and

C. Means for identifying any potential conflicts between a previously installed component and a component to be installed *(see Column 12, Lines 27-32).*

As per **Claim 15**, the Applicant appears to be attempting to invoke 35 U.S.C. 112, sixth paragraph, since it contains a "means-plus-function" limitation. However, the Examiner notes that the only "means" for performing the recited function in the specification appears to be computer program modules—software *per se*. While the claim meets the first prong of the three-prong analysis used to determine invocation of 35 U.S.C. 112, sixth paragraph, the claim does not meet the other prongs of the three-prong analysis, since no other specific corresponding structure or equivalents thereof are disclosed in the specification. Therefore, the claim limitation is not being treated under 35 U.S.C. 112, sixth paragraph.

Curtis discloses a system for installing software components **as in Claim 14 above**, and further discloses that the means for identifying components to be added to the system comprises means for accessing a semantic model comprising references among the components to be installed *(see Figure 5; Column 11, Lines 45-52; Column 13, Lines 7-27; and Column 14, Lines 42-45).*

As per **Claim 16**, the Applicant appears to be attempting to invoke 35 U.S.C. 112, sixth paragraph, since it contains a "means-plus-function" limitation. However, the Examiner notes that the only "means" for performing the recited function in the specification appears to be computer program modules—software *per se*. While the claim meets the first prong of the three-prong analysis used to determine invocation of 35 U.S.C. 112, sixth paragraph, the claim does not meet the other prongs of the three-prong analysis, since no other specific corresponding structure or equivalents thereof are disclosed in the specification. Therefore, the claim limitation is not being treated under 35 U.S.C. 112, sixth paragraph.

Curtis discloses a system for installing software components **as in Claim 15 above**, and further discloses that the system comprising means for loading an installation package including the semantic model *(see Figure 1, Element 17; Figure 2, Elements 101 and 220; Figure 5; Column 5, Lines 56-58; Column 7, Lines 40-45; and Column 13, Lines 7-9).*

As per **Claim 17**, Curtis discloses a system for installing software components **as in Claim 14 above**, and further discloses that the system comprising a data structure comprising references among the components to be installed *(see Figure 3; and Column 9, Lines 20-25).*

As per **Claim 18**, the Applicant appears to be attempting to invoke 35 U.S.C. 112, sixth paragraph, since it contains a "means-plus-function" limitation. However, the Examiner notes that the only "means" for performing the recited function in the specification appears to be computer program modules—software *per se*. While the claim meets the first prong of the three-prong analysis used to determine invocation of 35 U.S.C. 112, sixth paragraph, the claim does not meet the other prongs of the three-prong analysis, since no other specific corresponding structure or equivalents thereof are disclosed in the specification. Therefore, the claim limitation is not being treated under 35 U.S.C. 112, sixth paragraph.

Curtis discloses a system for installing software components **as in Claim 17 above**, and further discloses that the system comprising means for accessing the data structure *(see Column 9, Lines 20-25)*.

As per **Claim 19**, the Applicant appears to be attempting to invoke 35 U.S.C. 112, sixth paragraph, since it contains a "means-plus-function" limitation. However, the Examiner notes that the only "means" for performing the recited function in the specification appears to be computer program modules—software *per se*. While the claim meets the first prong of the three-prong analysis used to determine invocation of 35 U.S.C. 112, sixth paragraph, the claim does not meet the other prongs of the three-prong analysis, since no other specific corresponding structure or equivalents thereof are disclosed in the specification. Therefore, the claim limitation is not being treated under 35 U.S.C. 112, sixth paragraph.

Curtis discloses a system for installing software components **as in Claim 14 above**, and further discloses that the system comprising means for installing the components across a plurality of enterprise resources *(see Column 4, Lines 39-44; Column 5, Lines 52-55; and Column 8, Lines 53-60)*.

As per **Claim 20**, Curtis discloses a data structure associated with a software component installation package for identifying component incompatibilities, comprising:

A. An entry for each component previously installed on a system *(see Figure 3, Element 402; and Column 9, Lines 47-49)*; and

B. References associated with each previously component identifying any conflicting component to be installed *(see Column 10, Lines 20-35)*; whereby an alert is generated if an attempt is made to install a conflicting component *(see Column 10, Lines 36-40).*

As per **Claim 21,** Curtis discloses a computer program product of a computer-readable medium usable with a programmable computer, the computer program product having computer-readable code embodied therein for installing software components, the computer-readable code comprising instructions for:

A. Identifying components previously installed on a system *(see Column 10, Lines 60-64; and Column 11, Lines 11-20 and 57-61)*;

B. Identifying components to be installed on the system *(see Column 6, Lines 28-40 and 53-58; and Column 11, Lines 20-23)*; and

C. Identifying any potential conflicts between a previously installed component and a component to be installed *(see Column 12, Lines 27-32).*

As per **Claim 22,** Curtis discloses a computer program product for installing software components **as in Claim 21 above**, and further discloses that the instructions for identifying components previously installed on a system comprise instructions for accessing a semantic model comprising relationships among previously installed components *(see Figure 5; Column 11, Lines 45-52; Column 13, Lines 7-27; and Column 14, Lines 42-45).*

As per **Claim 23**, Curtis discloses a computer program product for installing software components **as in Claim 22 above**, and further discloses that the computer program product comprising instructions for updating the semantic model with the identity of newly installed components *(see Column 11, Lines 63-65; Column 12, Lines 66-67; and Column 13, Lines 28-47)*.

As per **Claim 24**, Curtis discloses a computer program product for installing software components **as in Claim 21 above**, and further discloses that the computer program product comprising instructions for providing a user with a plurality of options if a conflict is identified *(see Column 12, Lines 35-45)*.

As per **Claim 26**, Curtis discloses a computer program product for installing software components **as in Claim 24 above**, and further discloses that a second option includes continuing the installation *(see Column 12, Lines 51-53 and 57-59)*.

As per **Claim 27**, Curtis discloses a computer program product for installing software components **as in Claim 26 above**, and further discloses that the computer program product including instructions for, upon the exercise of the second option, recording an entry in a log indicative of the conflict and of the continuation of installation *(see Figure 2, Element 140; Column 6, Lines 65-67; Column 7, Lines 4-5; and Column 8, Lines 24-29)*.

As per **Claim 28,** Curtis discloses a computer program product for installing software components **as in Claim 21 above**, and further discloses that the computer program product comprising instructions for:

A. Initiating a removal of a component from the system *(see Figure 6, Element 560; and Column 13, Lines 50-51)*; and

B. Identifying remaining components which depend on the component to be removed *(see Column 13, Lines 59-62).*

As per **Claim 29,** Curtis discloses a computer program product for installing software components **as in Claim 28 above**, and further discloses that the computer program product comprising instructions for providing a user with a plurality of options if a dependent remaining component is identified *(see Column 12, Lines 35-45).*

As per **Claim 30,** Curtis discloses a computer program product for installing software components **as in Claim 29 above**, and further discloses that a first option includes aborting the removal *(see Figure 6, Element 570; and Column 13, Lines 62-63).*

As per **Claim 31,** Curtis discloses a computer program product for installing software components **as in Claim 29 above**, and further discloses that a second option includes continuing the removal *(see Figure 6, Element 568; and Column 13, Lines 55-56).*

As per **Claim 32**, Curtis discloses a computer program product for installing software components **as in Claim 28 above**, and further discloses that the computer program product comprising instructions for:

A. Identifying a first component previously installed on the system which is dependent upon a removed component *(see Column 13, Lines 4-6 and 64-67)*; and

B. Indicating the identity of a second component upon which the first component depends *(see Column 13, Lines 1-4).*

As per **Claim 33**, Curtis discloses a computer program product for installing software components **as in Claim 32 above**, and further discloses that the computer program product comprising instructions for:

A. Installing the second component upon which the first component depends *(see Column 13, Lines 1-4)*; and

B. Creating a dependency link between the first and second components *(see Column 13, Lines 28-47).*

As per **Claim 34**, Curtis discloses a method for installing software components, comprising:

A. Loading an installation package, the installation package including a component compatibility data structure *(see Figure 1, Element 17; Figure 2, Element 101; Figure 3, Element 400; Column 5, Lines 56-58; and Column 9, Lines 32-37)*;

B. Searching a target to which components are to be installed to identify installed components *(see Column 9, Lines 39-43)*;

C. Accessing the component compatibility data structure, the component compatibility data structure comprising, for each component A to be installed, a reference to any installed component with which the component A may conflict *(see Column 9, Lines 20-25)*; and

D. Determining whether a conflict is detected *(see Column 10, Lines 20-35)*.

As per **Claim 35,** Curtis discloses a method for installing software components **as in Claim 34 above**, and further discloses that the method comprising notifying a user of the conflict *(see Column 10, Lines 36-40)*.

As per **Claim 37,** Curtis discloses a method for installing software components **as in Claim 34 above**, and further discloses that the method comprising ignoring a detected conflict and continuing the installation *(see Column 12, Lines 35-45)*.

As per **Claim 38,** Curtis discloses a method for installing software components **as in Claim 37 above**, and further discloses that the method comprising entering in a log of the conflict *(see Figure 2, Element 140; Column 6, Lines 65-67; Column 7, Lines 4-5; and Column 8, Lines 24-29)*.

As per **Claim 39,** Curtis discloses a method for installing software components **as in Claim 34 above**, and further discloses that the method comprising:

A. Initiating the removal of an installed component *(see Figure 6, Element 560; and Column 13, Lines 50-51)*;

B. Accessing the component compatibility data structure *(see Column 9, Lines 20-25)*; and

C. Identifying a conflict if the installed component is removed *(see Column 10, Lines 20-35)*.

As per **Claim 40**, Curtis discloses a method for installing software components **as in Claim 34 above**, and further discloses that the method comprising:

A. Initiating an installation of a component B *(see Column 13, Lines 1-4)*;

B. Searching a target to which the component B is to be installed to identify installed components *(see Column 9, Lines 39-43)*;

C. Accessing the component compatibility data structure *(see Column 9, Lines 20-25)*; and

D. Determining if all of the components required by the component B are installed *(see Column 12, Lines 27-32)*.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. **Claims 5, 25, and 36** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Curtis (US 6,442,754)** in view of **Bourke-Dunphy et al. (US 6,918,112)**.

As per **Claim 5**, Curtis discloses a method for installing software components **as in Claim 4 above**. However, Curtis does not explicitly disclose that a first option includes aborting the installation.

In the same field of endeavor, Bourke-Dunphy et al. disclose a system and method for planning an installation procedure for a plurality of application and/or service components, where the user may reject the installation list by selecting a CANCEL action button to return to the component selection user interface to manually modify the component selections *(see Figure 5, Element 236; and Column 8, Lines 35-38)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to provide an option to abort the installation in the system of Curtis since there is already an option that allows the user to continue the installation. One would have been motivated to provide an option to abort the installation in order to mitigate the increased likelihood of errors occurring during component-based software installation by allowing the user to exit the current installation, correct the error identified, and reinitiate the installation procedure *(see Column 1, Lines 27-34).*

As per **Claim 25**, Curtis discloses a method for installing software components **as in Claim 24 above**. However, Curtis does not explicitly disclose that a first option includes aborting the installation.

In the same field of endeavor, Bourke-Dunphy et al. disclose a system and method for planning an installation procedure for a plurality of application and/or service components, where the user may reject the installation list by selecting a CANCEL action button to return to the component selection user interface to manually modify the component selections *(see Figure 5, Element 236; and Column 8, Lines 35-38).*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to provide an option to abort the installation in the system of Curtis since there is already an option that allows the user to continue the installation. One would have been motivated to provide an option to abort the installation in order to mitigate the increased likelihood of errors occurring during component-based software installation by allowing the user to exit the current installation, correct the error identified, and reinitiate the installation procedure *(see Column 1, Lines 27-34).*

As per **Claim 36**, Curtis discloses a method for installing software components **as in Claim 34 above**. However, Curtis does not explicitly disclose that the method further comprising aborting the installation if a conflict is detected.

In the same field of endeavor, Bourke-Dunphy et al. disclose a system and method for planning an installation procedure for a plurality of application and/or service components. The selected components and the added components are displayed as a list. When the selected components do not have a proper dependency, the interface further may identify the affected components and subcomponents. The user may then reject the installation list by selecting a CANCEL action button to return to the component selection user interface to manually modify the component selections *(see Figure 5, Element 236; and Column 8, Lines 19-30 and 35-38).*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to provide an option to abort the installation in the system of Curtis since there is already an option that allows the user to continue the installation. One would have been motivated to provide an option to abort the installation in order to mitigate the increased likelihood of errors occurring during component-based software installation by allowing the user to exit the current installation, correct the error identified, and reinitiate the installation procedure *(see Column 1, Lines 27-34).*

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

A. **Sonty et al. (US 5,499,357)** disclose a process for ensuring compatibility between components of a computer processing system, particularly for ensuring the integrity of a system configuration, even if enhancements to one system component render it incompatible with another component.

B. **Taylor (US 5,721,824)** discloses installing a software package having at least one dependent software package also to be installed on a server or standalone file space, multiple client file space or both in the file system of the server and one or more clients.

C. **Staelin (US 6,117,187)** discloses a method of automatically generating a software installation package that can be used to install an application program in a user's computer system.

D. **Davis (US 6,279,154)** discloses an apparatus and method for allowing a user to choose between typical and custom installation of third party application software packages on a computer system.

E. **Halpern et al. (US 6,282,711)** disclose methods used to customize the installation of software packages, as well as the installation of selected components of a software suite, on a local data processing system by downloading files from a remote server source coupled to a distributed processing network such as the Internet.

F. **Delo et al. (US 6,370,686)** disclose categorizing, accessing, and installing software components and presenting the location of these components to a requesting application.

G. **Forbes et al. (US 6,381,742)** disclose systems and methods for software distribution, and more particularly to the management of software packages after distribution.

H. **Zimniewicz et al. (US 6,618,857)** disclose a system and method for installing software on a computer prepares a computer system for the installation of a suite having a plurality of components.

I. **Marino et al. (US 6,681,391)** disclose a system and method for installing software on a computer generates an installation order that ensures that a component required for the functioning of another component is already installed.

J. **Te'eni et al. (US 6,725,452)** disclose a method for resolving dependency conflicts across diverse sets of functional entities while installing or removing specific operative elements in a computing environment.

K. **Keller et al. (US 6,847,970)** disclose methods and apparatus for managing dependencies among the various components of distributed computing systems.

L. **Huang et al. (US 6,968,539)** disclose methods and apparatus for processing of web applications that are written in the form of web pages, which can be downloaded through the Internet using web communication protocols, installed in local computers, and executed utilizing web components, such as a web browser and JavaScript™ interpreter, in the local computers.

M. **Newman (US 6,983,449)** discloses a system and method for configuring software for distribution that identifies conflicts between two or more components in a software suite and resolves the conflicts before distributing the software.

N. **Cicciarelli et al. (US 2002/0188941)** disclose systems, methods, and computer program products for improving the installation of software application packages by using an incremental conditional installation process (and, optionally, caching of selected install components).

O. **Jhanwar et al. (US 2004/0093593)** disclose facilitating creation of new software products based on the componentized software product and servicing the software product.

P. **Kaminsky et al. (US 2004/0117783)** disclose dependency management during the installation of upgraded application components.

Q. **Mullen et al. (US 2004/0243997)** disclose a system, method, and program for installing program components on a computer.

R. **Sierer et al. (US 2004/0255291)** disclose a system and method for programmatically generating an application system installer using component dependency analysis, where the installer is operable to deploy an application system onto a target system to perform a desired function.

S. **Klinedinst et al. (EP 1096374)** disclose a system and method that provide initial installation of one or more software components for a self-service financial transaction (SSFT) terminal that is coupled with a global communications network.

T. **Paton et al. (GB 2394570)** disclose a method for installing software components and optimizing such installation.

U. **Hoek et al. ("Software Release Management")** disclose a tool called SRM (Software Release Manager) and process through which software is made available to and obtained by its users.

V. **Carzaniga et al. ("A Characterization Framework for Software Deployment Technologies")** disclose a framework for characterizing existing and proposed deployment technologies.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The

Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, James W. Myhre, can be reached on 571-270-1065. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / ac
June 9, 2006

James W. Myhre
Supervisory Patent Examiner